Evolutionary Multiobjective Optimization Based Multimodal Optimization: Fitness Landscape Approximation and Peak Detection

Cheng, Ran; Li, Miqing; Li, Ke; Yao, Xin

DOI:
10.1109/TEVC.2017.2744328

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Cheng, R, Li, M, Li, K & Yao, X 2018, 'Evolutionary Multiobjective Optimization Based Multimodal Optimization: Fitness Landscape Approximation and Peak Detection', *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 692 - 706. https://doi.org/10.1109/TEVC.2017.2744328

Evolutionary Multiobjective Optimization-Based Multimodal Optimization: Fitness Landscape Approximation and Peak Detection

Ran Cheng, *Member, IEEE*, Miqing Li, Ke Li, *Member, IEEE*, and Xin Yao, *Fellow, IEEE*

***Abstract*—Recently, by taking advantage of evolutionary multiobjective optimization techniques in diversity preservation, the means of *multiobjectivization* has attracted increasing interest in the studies of multimodal optimization (MMO). While most existing work of multiobjectivization aims to find all optimal solutions simultaneously, in this paper, we propose to approximate multimodal fitness landscapes via multiobjectivization, thus providing an estimation of potential optimal areas. To begin with, an MMO problem is transformed into a multiobjective optimization problem (MOP) by adding an adaptive diversity indicator as the second optimization objective, and an approximate fitness landscape is obtained via optimization of the transformed MOP using a multiobjective evolutionary algorithm. Then, on the basis of the approximate fitness landscape, an adaptive peak detection method is proposed to find peaks where optimal solutions may exist. Finally, local search is performed inside the detected peaks on the approximate fitness landscape. To assess the performance of the proposed algorithm, extensive experiments are conducted on 20 multimodal test functions, in comparison with three state-of-the-art algorithms for MMO. Experimental results demonstrate that the proposed algorithm not only shows promising performance in benchmark comparisons, but also has good potential in assisting preference-based decision-making in MMO.**

***Index Terms*—Decision-making, fitness landscape approximation, multimodal optimization (MMO), multiobjective optimization, multiobjectivization, niching, peak detection, preference.**

Manuscript received October 17, 2016; revised February 13, 2017, June 12, 2017, and August 16, 2017; accepted August 22, 2017. Date of publication September 15, 2017; date of current version September 28, 2018. This work was supported by the Engineering and Physical Sciences Research Council under Grant EP/K001523/1 and Grant EP/J017515/1. The work of X. Yao was supported by a Royal Society Wolfson Research Merit Award. *(Corresponding author: Xin Yao.)*

R. Cheng and M. Li are with the Center of Excellence for Research in Computational Intelligence and Applications, School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: ranchengcn@gmail.com; limitsing@gmail.com; keli.genius@gmail.com).

K. Li is with the Department of Computer Science, University of Exeter, Exeter EX4 4QF, U.K. (e-mail: k.li@exeter.ac.uk).

X. Yao is with the Center of Excellence for Research in Computational Intelligence and Applications, School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K., and also with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: x.yao@cs.bham.ac.uk).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2017.2744328

I. Introduction

MULTIMODAL optimization (MMO), which refers to single-objective optimization involving multiple optimal (or near-optimal) solutions, has attracted increasing interest recently [1]–[3]. MMO is widely seen in real-world scenarios, where the decision-makings can be made on the basis of multiple optimal solutions of a given optimization problem [4]. For example, in truss-structure optimization [5], where the optimization objective is the quality criterion (e.g., weight or reliability) of the truss structure and the decision variables can be the density or length of the truss members, it is likely that different values of the decision variables can lead to the same (or very close) fitness of the objective function. In such a scenario, the decision maker (DM) has to make decisions according to personal preferences. There are also many other real-world applications of MMO as reviewed in [4], such as virtual camera composition [6], metabolic network modeling [7], laser pulse shaping [8], job scheduling [9], [10], data clustering [11], feature selection [12], and neutral network ensembles [13].

In MMO, since there exist more than one optimal solution to be found simultaneously, population-based metaheuristics such as evolutionary algorithms (EAs) provide a suitable solution framework, which maintains a set of candidate solutions during one single run. However, since most EAs have been originally designed for conventional single-objective optimization which involves only one optimal solution, they are not directly applicable to MMO due to their poor capability of population diversity preservation [14]. To address such an issue, researchers have proposed a variety of solution approaches that can be roughly categorized into the following three groups.

The first group is known as the *niching* approaches [15], where the basic idea is to adaptively preserve diverse subpopulations converging toward different optimal solutions for local exploitations. Some early work along this direction includes the clearing procedures [16], [17], the crowding techniques [18], [19], the sharing methods [20]–[22], the clustering-based schemes [23], [24], the restricted tournament selection strategies [25], [26], and the speciation techniques [27], [28]. However, since most of early niching approaches are designed on the basis of threshold parameters such as crowding size and niching radius, their performance is often sensitive to parameter settings. Therefore, most recent work has been focused on adaptive/parameterless

niching approaches. For example, a recursive middling sampling approach has been proposed to continuously sample the fitness landscape until a predefined termination condition is satisfied [29]; a topological species conservation strategy has been proposed to avoid extinction of some niches by means of a seed preservation method [30]. More recently, novel clustering-based niching methods have also been proposed to transform sensitive parameters (e.g., crowding size) to a less sensitive parameter as cluster size [31], [32].

The second group aims to enhance population diversity by adopting novel reproduction/update operators, where the motivation is to modify conventional single-objective population-based metaheuristics such as particle swarm optimization (PSO) [33] and differential evolution (DE) [34] for MMO. Among some representative work, Qu *et al.* [35] proposed a locally informed PSO algorithm, where multiple local best positions are used to guide search of each particle to converge to different optimal subspaces. Fieldsend [36] proposed a localized EA using Gaussian process-based local surrogate models, where training and sampling of the models are performed inside the dynamically detected niche peaks. Biswas *et al.* [37] proposed two different reproduction operators for two types of candidate solutions in a local informative DE algorithm; and most recently, Yang *et al.* [38] proposed a multimodal ant colony optimization algorithm based on a novel adaptively local search operator.

Recently, some attempts have been made to transform an MMO problem (MMOP) into a multiobjective optimization problem (MOP) [39], a process known as the *multiobjectivization* [40]. Usually, such a transformed MOP consists of two objectives: the first objective is the given MMOP, and the second objective is a diversity indicator constructed based on either gradient information [29], [41] or distance information of each candidate solution [42], [43]. In a more recent study, Wang *et al.* [44] pointed out that the conflicts between objectives of the transformed MOP play an important role in a successful multiobjectivization approach. They proposed a novel transformation method to reconstruct both objectives.

Compared to conventional niching or diversity enhancement approaches, the multiobjectivization approaches have two major advantages. First, once an MMOP is properly transformed to an MOP, existing multiobjective EAs (MOEAs) [45] can be applied to the transformed MOP with few additional modifications, thus saving efforts in designing new algorithms. Second, since the objectives of the transformed MOP are designed to be in conflict with each other (i.e., convergence versus diversity), an implicit niching effect can be achieved without cumbersome tunings of problem-dependent parameters. By taking these advantages of multiobjectivization, in this paper, we propose a new evolutionary multiobjective optimization-based MMO (EMO-MMO) algorithm. Unlike most existing multiobjectivization approaches which aim to locate all optimal solutions simultaneously, the proposed algorithm first performs explorations to obtain an approximate fitness landscape by archiving the candidate solutions obtained during the EMO process. Then, with the approximate fitness landscape, a peak detection method is designed to locate peaks where optimal solutions may exist. And finally, a local optimizer is used to perform exploitations inside each located peak to obtain the final optimal solutions. The main contributions of this paper can be summarized as follows.

1) A general algorithm framework of EMO-MMO is proposed, which consists of three components: a) fitness landscape approximation; b) peak detection; and c) local search. On one hand, the proposed EMO-MMO can be used to perform general optimization to obtain multiple optimal solutions, and on the other hand, it can be also used to obtain approximate fitness landscapes to assist preference-based decision-making.
2) In order to obtain approximate fitness landscapes, a multiobjective fitness landscape approximation (MOFLA) method is proposed. In the proposed MOFLA, a given MMOP is first transformed to an MOP by adding a diversity indicator as the second optimization objective. Considering that the requirement of population diversity may dynamically change during the optimization process, the diversity indicator is designed to be adaptively related to the number of generations. In addition, to achieve a better balance between convergence and diversity, a discrete grid coordinate system is adopted instead of the original continuous coordinate system in the proposed diversity indicator. An MOEA is applied to the optimization of the transformed MOP, and the candidate solutions obtained during the optimization process are archived as the approximate fitness landscape.
3) In order to perform decision-making using the approximate multimodal fitness landscape, an adaptive peak detection (APD) method is proposed to locate promising peaks where optimal solutions may exist. The proposed method performs binary cuttings on the approximate fitness landscape and tries to locate all promising peaks on each cutting slice. Empirical results demonstrate that the proposed peak detection method, without cumbersome parameter tunings, performs robustly on a variety of approximate fitness landscapes.

The rest of this paper is organized as follows. Section II presents some background knowledge of EMO-MMO, together with some discussion on the relationship between the two topics. Afterwards, based on the discussion, motivations of this paper are further illustrated. Section III details the proposed EMO-MMO, including the algorithm framework, the MOFLA method and the peak detection method. Experimental study is presented in Section IV. We first conduct some comparisons with three state-of-the-art algorithms for MMO. Then, performance of the proposed MOFLA method and peak detection method is further assessed. Finally, Section VI draws the conclusion.

II. Background

A. Evolutionary Multiobjective Optimization

MOPs, which involve more than one conflicting objective to be optimized simultaneously, can be briefly

formulated as follows:[1]

$$\begin{aligned} \text{maximize} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\ \text{s.t.} \quad & \mathbf{x} \in X, \quad \mathbf{f} \in Y \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_D) \in X$ denotes a decision vector in decision space $X \subseteq \mathbb{R}^D$, $\mathbf{f} \in Y$ denotes a objective vector in objective space $Y \subseteq \mathbb{R}^M$, and D and M denote the number of decision variables and the number of objectives, respectively. Given a set of box constraints, the decision space X can be presented as

$$X = \prod_{i=1}^{D} [L_i, U_i] \tag{2}$$

where L_i and U_i denote the lower and upper boundaries for each decision variable x_i, respectively.

Since there exist conflicts between the optimization objectives $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})$ in an MOP as formulated above, it is impossible to find one single solution that optimizes all objectives simultaneously. Instead, a set of optimal solutions, known as Pareto optimal solutions, can be obtained to represent the tradeoffs between different objectives. To be specific, given two candidate solutions $\mathbf{x}_1$ and $\mathbf{x}_2$, solution $\mathbf{x}_1$ is said to *dominate* the other solution $\mathbf{x}_2$ iff

$$\begin{cases} \forall i \in 1, 2, \ldots, M : f_i(\mathbf{x}_1) \geq f_i(\mathbf{x}_2) \\ \exists j \in 1, 2, \ldots, M : f_j(\mathbf{x}_1) > f_j(\mathbf{x}_2). \end{cases} \tag{3}$$

If a solution $\mathbf{x}^*$ cannot be dominated by any other solutions in X, then $\mathbf{x}^*$ is known as *Pareto optimal*, and the union of all $\mathbf{x}^*$ is known as the Pareto set (PS), while the image of PS in the objective space, namely, the union of $\mathbf{f}(\mathbf{x}^*)$, is known as the Pareto front (PF). In order to approximate the PF (or PS), a variety of MOEAs have been proposed during the past two decades [45].

B. Multimodal Optimization

MMOPs, which involve multiple global optimal solutions of a single objective to be obtained simultaneously, can be formulated as follows:

$$\begin{aligned} \text{maximize} \quad & g(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in X \end{aligned} \tag{4}$$

where $g(\mathbf{x})$ is the objective function, and $\mathbf{x} = (x_1, \ldots, x_D) \in X$ is the decision vector.

Given an MMOP as formulated in (4), there exist a set of global optimal solutions X^* that maximize the objective function $f(\mathbf{x})$ as

$$X^* = \{\mathbf{x} \in X : \{\nexists \mathbf{y} \in X : f(\mathbf{y}) > f(\mathbf{x}) \wedge \mathbf{y} \neq \mathbf{x}\}\} \tag{5}$$

where $|X^*| > 1$ holds. Specifically, this paper only considers MMOPs having a finite number of discretely distributed global optimal solutions, namely, where X^* is a finite set. By contrast, for MMOPs having an infinite set of continuously distributed optimal solutions, some further related discussion are given in Section V-C.

[1]Without loss of generality, this paper only considers maximization problems. Minimization problems can be equivalently transformed to maximization problems by taking negative values of the objective function.

C. Transformation From MMOPs to MOPs

In order to apply EMO techniques to MMO, most existing approaches try to transform an MMOP into an MOP by introducing a diversity indicator as an additional optimization objective, while the optimization objective of the original MMOP remains unchanged

$$\begin{aligned} &\text{maximize } \mathbf{f}(\mathbf{x}) = (g(\mathbf{x}), d(\mathbf{x})) \\ &\text{s.t. } \mathbf{x} \in X, \quad \mathbf{f} \in Y \end{aligned} \tag{6}$$

where $g(\mathbf{x})$ is the objective function of an MMOP as formulated in (4), and $d(\mathbf{x})$ is an indicator that measures the diversity of decision vector $\mathbf{x}$ of a candidate solution. To construct the diversity indicator $d(\mathbf{x})$, most approaches make use of gradient or distance related information, and some of the representatives are as follows.

As an early representative work using gradient information, Yao *et al.* [29] proposed to make use of the absolute value of the gradient of $g(\mathbf{x})$ to construct the diversity indicator

$$d_1(\mathbf{x}) = \frac{\sum_{i=1}^{D} |\frac{\partial g}{\partial x_i}|}{D}. \tag{7}$$

In addition to the first-order gradient, Deb and Saha [41] have also attempted to use the second-order gradient information to avoid the scenario of weak Pareto optimality

$$d_2(\mathbf{x}) = |g'(\mathbf{x})| + \left(1 - \text{sign}\left(g''(x)\right)\right) \tag{8}$$

where sign$(\cdot)$ returns $+1$ and -1 for positive and negative operands, respectively.

Considering that gradient information may not always be available in practice, some researchers proposed to use distance-based information to construct the diversity indicator. For example, Basak *et al.* [43] proposed to use the mean distance from each candidate solution to the others

$$d_3(\mathbf{x}) = \frac{\sum_{j=1}^{N} \|\mathbf{x} - \mathbf{x}_j\|}{N} \tag{9}$$

where N is the number of candidate solutions in the population. Similarly, Bandaru and Deb [42] proposed to use such a distance-based diversity indicator as the second objective in their niching NSGA-II algorithm.

Since the most elementary characteristic of an MOP is the conflicting nature between different objectives, it is important that $d(\mathbf{x})$ is designed to be in conflict with the original objective $g(\mathbf{x})$, such that MOEAs are able to work properly. To address such an important issue, Wang *et al.* [44] proposed to modify both objectives to guarantee the conflicts between them.

D. Motivations

As presented above, since the target of both EMO and MMO is to obtain a set of equally important optimal solutions, the motivation in the design of EMO/MMO algorithms shares substantial similarity: in EMO, a set of candidate solutions are obtained as an approximation to the true PF, which will require that the candidate solutions are not only evenly distributed but also as close to the true PF as possible; in MMO, similarly, there also exist a set of optimal solutions to be found

Algorithm 1 Main Framework of EMO-MMO

1: **Input:** the maximum number of generations t_{max}, the MMOP to be optimized $g(\mathbf{x})$;
2: **Output:** optimal solution set $\mathcal{S}$;
3: /*Multiobjective Fitness Landscape Approximation*/
4: $\mathcal{D} = MOFLA(t_{max}, g(\mathbf{x}))$;
5: /*Peak Detection*/
6: $\mathcal{P} = PeakDetection(\mathcal{D})$;
7: /*Local Search*/
8: $\mathcal{S} = LocalSearch(g(\mathbf{x}), \mathcal{P})$;

simultaneously, which have the same (or very similar) fitness value. Therefore, a successful EMO/MMO algorithm should strike a good balance between convergence and diversity of the population.

However, most existing MOEAs are not directly applicable to the optimization of MMOPs due to the fact that MMO has more strict requirement of population diversity than EMO. In multiobjective optimization, since it can be deduced from the Karush–Kuhn–Tucker optimality conditions that the PF (as well as PS) is a piecewise continuous manifold [46], [47], there often exists a strong regularity between the candidate solutions close to the PF. In MMO, however, there is no such regularity property that can be taken advantage of. By contrast, the multiple global optimal solutions can be sparsely distributed in different locations of the fitness landscape with little correlation. Therefore, as pointed out in [40], if the target is to obtain a set of relative good solutions (instead of all accurate optimal solutions), EMO techniques can be used to perform wide explorations in the multimodal fitness landscapes, although the accuracy of the optimal solutions cannot be guaranteed.

In this paper, we propose a new EMO-MMO algorithm, where an MOFLA method is designed on the basis of an MOEA. To transform an MMOP to an MOP for deploying the proposed MOFLA method, a diversity indicator is designed to be the second objective of the transformed MOP. In spite of existing indicators as given in Section II-C, all of them are constructed with a fixed formulation. In practice, however, the required balance between convergence and diversity can dynamically change as optimization proceeds. To address this issue, we propose an adaptive diversity indicator which is related to the number of generations, thus striking an adaptive balance between convergence and diversity during the optimization process.

To make use of the approximate fitness landscape, an APD method is proposed to find promising peaks where optimal solutions may exist. And finally, based on the approximate fitness landscape together with the detected peaks, independent local search can be further performed inside each peak to exploit for the final optimal solutions.

III. Proposed Algorithm

A. Framework

The main framework of the proposed EMO-MMO is summarized in Algorithm 1, from which we can see that

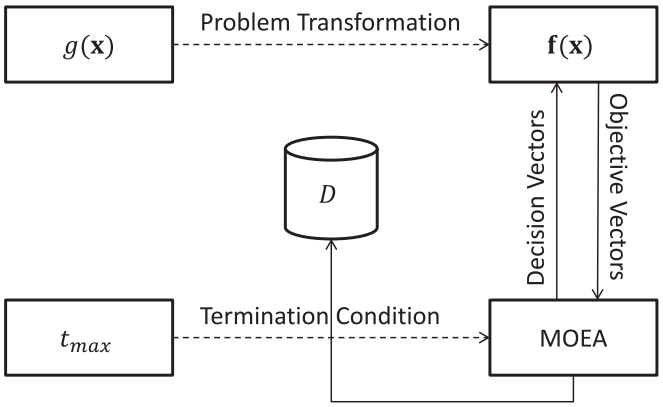


Fig. 1. Framework of the MOFLA component. $g(\mathbf{x})$ and $t_{\max}$ are two inputs of this framework, where $g(\mathbf{x})$ is the MMOP to be optimized and $t_{\max}$ is the maximum number of generations as the termination condition. To deploy MOFLA, the given MMOP $g(\mathbf{x})$ is first transformed to an MOP $\mathbf{f}(\mathbf{x})$, and an existing MOEA is applied to the optimization of the transformed MOP. By archiving the candidate solutions created during the multiobjective optimization process, $\mathcal{D}$ stores the approximate fitness landscape.

EMO-MMO consists of three main components: 1) MOFLA; 2) peak detection; and 3) local search. In MOFLA, the given MMOP is transformed to an MOP, and an MOEA is applied on the transformed MOP to approximate the multimodal fitness landscape; then, with the approximate fitness landscape, a peak detection method is used to find out all potential peaks where optimal solutions may exist; and finally, local search is performed inside each detected peak. The following sections will detail the three main components in Algorithm 1 successively.

B. Multiobjective Fitness Landscape Approximation

As illustrated in Fig. 1, the MOFLA component further consists of two subcomponents: a transformed MOP $\mathbf{f}(\mathbf{x})$ and an MOEA. In addition, there are two inputs, one of which is the MMOP to be optimized, denoted as $g(\mathbf{x})$, and the other is the maximum number of generations $t_{\max}$ as the termination condition. As the approximate fitness landscape, the candidate solutions generated during the multiobjective optimization process are stored in an external archive $\mathcal{D}$.

At the first step of MOFLA, the given MMOP is first transformed to an MOP as formulated in (6). To be specific, the given, MMOP $g(\mathbf{x}_{t,i})$, still remains unchanged as the first objective function in the transformed MOP, where $\mathbf{x}_{t,i} = (x_{t,i,1}, \ldots, x_{t,i,D})$ denotes a decision vector in the population P_t of generation t; while for the second objective $d(\mathbf{x}_{t,i})$, we adopt a grid-based diversity indicator, which is inspired from the grid-based techniques widely applied in the EMO community for diversity management [48]–[53].

In the proposed grid-based diversity indicator, each decision variable value $x_{t,i,j}$ is normalized using a discrete grid coordinate system as

$$x'_{t,i,j} = \left\lfloor (N-1) \times \left(\frac{x_{t,i,j} - x_{t,j}^{\min}}{x_{t,j}^{\max} - x_{t,j}^{\min}} \right) \right\rfloor + 1 \qquad (10)$$

where $x'_{t,i,j}$ denotes the new decision variable value inside the grid coordinate system, $x^{\max}_{t,j}$ and $x^{\min}_{t,j}$ are the upper and lower boundaries of the jth decision variable estimated using all decision vectors in population P_t, and $N = |P_t|$ is the population size. With such a grid-based normalization strategy, each dimension of the decision space is divided into a number of N hyperboxes, and all decision variable values can be consequently truncated into discrete values of $\{1, 2, \ldots, N\}$. As a consequence, given a number of N candidate solutions, it is expected that there will be at most one candidate solution inside each hyperbox in the extreme cases, thus maximizing the potential diversity measurement capability of the grid coordinate system.

Once the decision variable values of $\mathbf{x}_{t,i}$ are normalized using the grid coordinate system into $\mathbf{x}'_{t,i}$, the diversity quality of $\mathbf{x}_{t,i}$ can be measured on the basis of the Manhattan distances (L_1 norm) between $\mathbf{x}'_{t,i}$ and all the other decision vectors in the niche (namely, neighborhood) it belongs to

$$d_{\text{grid}}(\mathbf{x}_{t,i}) = \frac{1}{\delta_t}\left(\sum_{k\in K_{t,i}} \|\mathbf{x}'_{t,i} - \mathbf{x}'_{t,k}\|_1\right) - |K_{t,i}| \tag{11}$$

where $K_{t,i}$ contains the indices for the decision vectors in the niche that $\mathbf{x}'_{t,i}$ belongs to, defined as

$$K_{t,i} = \left\{j \in \{1, \ldots, N\} : \|\mathbf{x}'_{t,i} - \mathbf{x}'_{t,j}\|_1 < \delta_t\right\} \tag{12}$$

with δ_t being an adaptive niche radius

$$\delta_t = \left(1 - \frac{t-1}{t_{\max}}\right) \times \max_i\left\{\min_{j\neq i}\|\mathbf{x}'_{t,i} - \mathbf{x}'_{t,j}\|_1\right\}. \tag{13}$$

It can be seen that the grid-based diversity indicator $d_{\text{grid}}(\mathbf{x}_{t,i})$ consists of two parts. The first part, which sums up the normalized Manhattan distances from $\mathbf{x}'_{t,i}$ to all the others inside the niche defined by the adaptive niche radius δ_t, is used to measure the local distribution of the decision vectors. As a consequence, inside each niche, the more sparsely the decision vectors are distributed, the larger the summed up distance will be. By contrast, the second part, $|K_{t,i}|$, which is the total number of decision vectors inside each niche (i.e., niche count), is another important measurement to reflect the local density of the decision vectors. Correspondingly, a smaller $|K_{t,i}|$ indicates better population diversity and vice versa.

As another important factor in the proposed MOFLA method, the adaptive niche radius δ_t is designed out of the following considerations. First, due to the different requirements of balance between convergence and diversity in different phases of multiobjective optimization, it will be more beneficial if the diversity indicator is related to the number of generations t. Therefore, the coefficient $(1 - [(t-1)/t_{\max}])$ is used to linearly reduce the niche radius, such that increasing emphasis on convergence can be exerted in the late optimization phase. It is worth noting that $(1 - [(t-1)/t_{\max}])$ can be also generalized into $(1 - [(t-1)/t_{\max}])^{\alpha}$, such that setting the values of α will generate different changing rates of the coefficient. However, as indicated by our empirical results summarized in Section V in the supplementary material, on one hand, the indicator is not particularly sensitive to the changing rate of the coefficient as long as it is reduced mildly with the increase of t; on the other hand, if the coefficient becomes constant by setting α to 0, the performance of the algorithm has a significant deterioration on the problems with a large number of densely distributed optimal solutions (i.e., f_8 and f_9 as presented in Section V in the supplementary material). Therefore, we directly adopt the linear changing rate in this paper for simplicity.

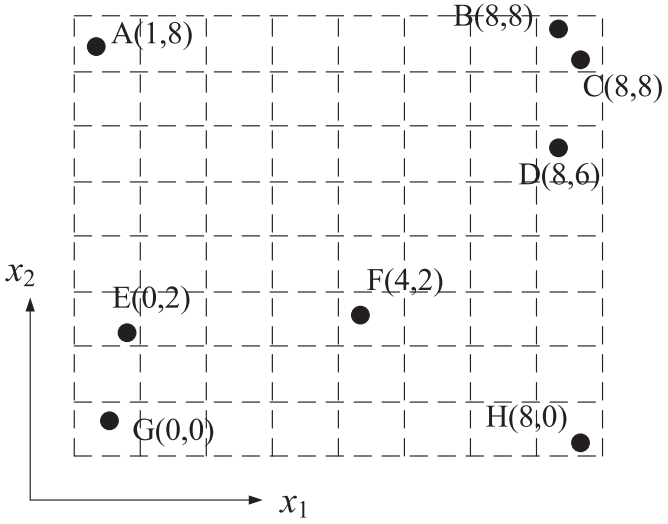


Fig. 2. Illustrative example of the proposed grid-based diversity indicator in a 2-D decision space. Calculated by (10)–(13) (assuming $\delta_t = 5.0$) in Section III-B, the diversity indicator values d_{grid} of the decision vectors A–H are -1.0, -2.6, -2.6, -2.2, -1.8, -1.2, -1.6, and -1.0, respectively.

Second, in practice, due to the various shapes of different peaks, it is difficult to determine a fixed niche radius for generic usage without priori knowledge about the MMOP to be solved. Therefore, the niche radius is adaptively estimated on the basis of the distances between the neighboring decision vectors in each generation, where the maximum neighboring distance is used as the largest possible threshold for the niche radius, as formulated in (13).

As further illustrations to the proposed grid-based diversity indicator, a schematic diagram is given in Fig. 2. To be specific, we have the following observations. First, given a decision vector, the maximum possible diversity value (-1.0) means that there is no other neighbor in its niche, such as A and H in this example. Second, since the diversity value of a decision vector is determined by the number of its neighbors and the distances between it and these neighbors, decision vectors having more neighbors or closer distances to their neighbors are likely to obtain smaller diversity values. For example, E has a smaller diversity value than F because E has one more neighbor than F; C has a smaller diversity value than E because the distances of C to its neighbors are shorter than those of E to its neighbors, even though C and E have the same number of neighbors. Third, for decision vectors such as B and C which are inside the same hyperbox, they have the same diversity value. It means that if two decision vectors are too close to each other (i.e., inside the same hyperbox), they are considered to have the same contribution to the population diversity, thus to be further distinguished by the objective

Algorithm 2 NSGA-II-Based MOFLA

1: **Input:** the maximum number of fitness evaluations t_{max}, population size N, the MMOP to be optimized $g(\mathbf{x})$;
2: **Output:** the approximate fitness landscape $\mathcal{D}$;
3: **Initialization:** create the initial population $P_0 = (X_0, Y_0)$ with N randomized individuals, where X_0 and Y_0 contain the decision vectors and objective vectors respectively, set $\mathcal{D} = P_0$ and $t = 0$;
4: /*Main Loop*/
5: **while** $t < t_{max}$ **do**
6: /*Reproduction*/
7: $\bar{X}_t$ = recombination+mutation(X_t): perform simulated binary crossover and polynomial mutation;
8: $\bar{Y}_t$ = evaluation($\bar{X}_t \cup X_t$): evaluate the merged decision vector set using the transformed MOP in (14);
9: $P_t = (\bar{X}_t, \bar{Y}_t)$;
10: /*Selection*/
11: $(F_{t,1}, F_{t,2}, ..., F_{t,l})$ = non-dominated-sorting(P_t): perform non-dominated sorting on P_t to divide the population into a number of non-dominated fronts $F_{t,1}, F_{t,2}, ...$, and select candidate solutions successively until front F_l is reached such that $|F_{t,1}| + |F_{t,2}| + ... + |F_{t,l}| \geq N$ and $|F_{t,1}| + |F_{t,2}| + ... + |F_{t,l-1}| < N$;
12: S_t = crowding-distance-assignment($F_{t,l}$): calculate the crowding distance for each candidate solution in $F_{t,l}$, and select a number of $N-(|F_{t,1}|+|F_{t,2}|+...+|F_{t,l-1}|)$ candidate solutions from F_l which have minimal crowding distances;
13: $P_{t+1} = \{F_{t,1}, F_{t,2}, ..., F_{t,l-1}\} \cup S_t$;
14: /*Archiving*/
15: $\mathcal{D} = \mathcal{D} \cup P_{t+1}$;
16: $t = t + 1$;
17: **end while**

function $g(\mathbf{x})$ of the original MMOP. In terms of the effectiveness of the grid coordinate system, some empirical discussion can be found in Section V-A.

With the grid-based diversity indicator as formulated in (10)–(14), an MMOP can now be transformed into the following MOP:

$$\mathbf{f}_{\text{grid}}(\mathbf{x}) = \big(g(\mathbf{x}), d_{\text{grid}}(\mathbf{x})\big) \tag{14}$$

where $\mathbf{f}_{\text{grid}}(\mathbf{x})$ denotes the transformed MOP, and $g(\mathbf{x})$ and $d_{\text{grid}}(\mathbf{x})$ denote the original MMOP and the grid-based diversity indicator, respectively. Once an MMOP is transformed into $\mathbf{f}_{\text{grid}}(\mathbf{x})$ as above, an existing MOEA can be directly applied to perform multiobjective optimization on it. Here, as an example, we present how to apply one of the most classic MOEAs, namely, the NSGA-II [54], to the optimization of the transformed MOP $\mathbf{f}_{\text{grid}}(\mathbf{x})$. Other MOEAs can also be applied in a similar way.

As presented in Algorithm 2, the NSGA-II-based MOFLA has a very similar framework as original NSGA-II, except that the population created in each generation has been stored in an external archive $\mathcal{D}$ as an approximation to the fitness landscape. As pointed out in a recent study [55], using large archives to store historical candidate solutions is particularly beneficial in capturing the topological structures of multimodal fitness landscapes. Therefore, in the proposed MOFLA, we also store all historical candidate solutions in $\mathcal{D}$ as an approximation of the fitness landscape. Despite that archiving all candidate solutions requires some additional memory space, it provides useful information such as the positions of peaks where optimal solutions could exist. As will be presented in the following section, the proposed peak detection method works properly on the basis of $\mathcal{D}$ without costing any additional fitness evaluations (FEs). Moreover, once the peaks are detected, the DM will be able to perform further exploitations merely inside the regions of interest (ROIs). This is particularly desirable when the FEs are computationally expensive.

One thing to be noted is that, at step 8, the offspring decision vector set $\bar{X}_t$ should be merged with the parent decision vector set X_t before performing FEs. This is due to the fact that the calculation of diversity indicator $d_{\text{grid}}(\mathbf{x})$ should be conducted on $\bar{X}_t \cup X_t$ (instead of $\bar{X}_t$ or X_t alone), such that the diversity indicator values are synchronized based on the topology of the merged population. For the $g(\mathbf{x})$ function (fitness) values of X_t, which still remain unchanged, are directly copied to $\bar{Y}_t$ to save redundant FEs.

C. Peak Detection

Once the approximate fitness landscape $\mathcal{D}$ is generated by Algorithm 2, we shall conduct further analyses to mine useful information from it. Since $\mathcal{D}$ is an approximate fitness landscape, it does not directly indicate the exact positions of the optimal solutions. Nevertheless, it is also beneficial to know the potential subregions of the fitness landscape where optimal solutions may exist, such that local search can be further performed in each of them independently. In such an optimal subregion, all the other solutions around the optimal solution should have relatively inferior fitness values, thus naturally forming a *peak* in the fitness landscape

$$\psi = \big(X_\psi, Y_\psi\big) : \begin{cases} X_\psi \subseteq X \\ Y_\psi = \big\{f(\mathbf{x}) : \mathbf{x} \in X_\psi\big\} \end{cases} \tag{15}$$

satisfying

$$\exists \mathbf{x}^*_\psi \in X_\psi : \Big\{\forall \mathbf{x} \in X_\psi \backslash \Big\{\mathbf{x}^*_\psi\Big\} : f(\mathbf{x}) < f\big(\mathbf{x}^*\big)\Big\} \tag{16}$$

where X is the entire feasible decision space, X_ψ denotes the region in the decision space covered by the peak, Y_ψ contains the fitness values (i.e., peak heights) in correspondence with the decision vectors in X_ψ, and $\mathbf{x}^*_\psi$ is the optimal solution inside the peak region specified by ψ.

Since there is only one optimal solution in each peak as defined by (15) and (16), once the peaks are located, local search can be performed in a parallel manner inside each peak to exploit the corresponding optimal solution, which will substantially increase the concurrency of the optimization process. In addition, since a DM may only be interested in part (but not all) of the peaks, it will also save a lot of FEs by exploiting specific peaks according to the DM's preferences.

Despite the fact that peaks provide very useful information of a multimodal landscape, it is difficult to obtain their specific locations in practice. For example, as shown in Fig. 3(a),

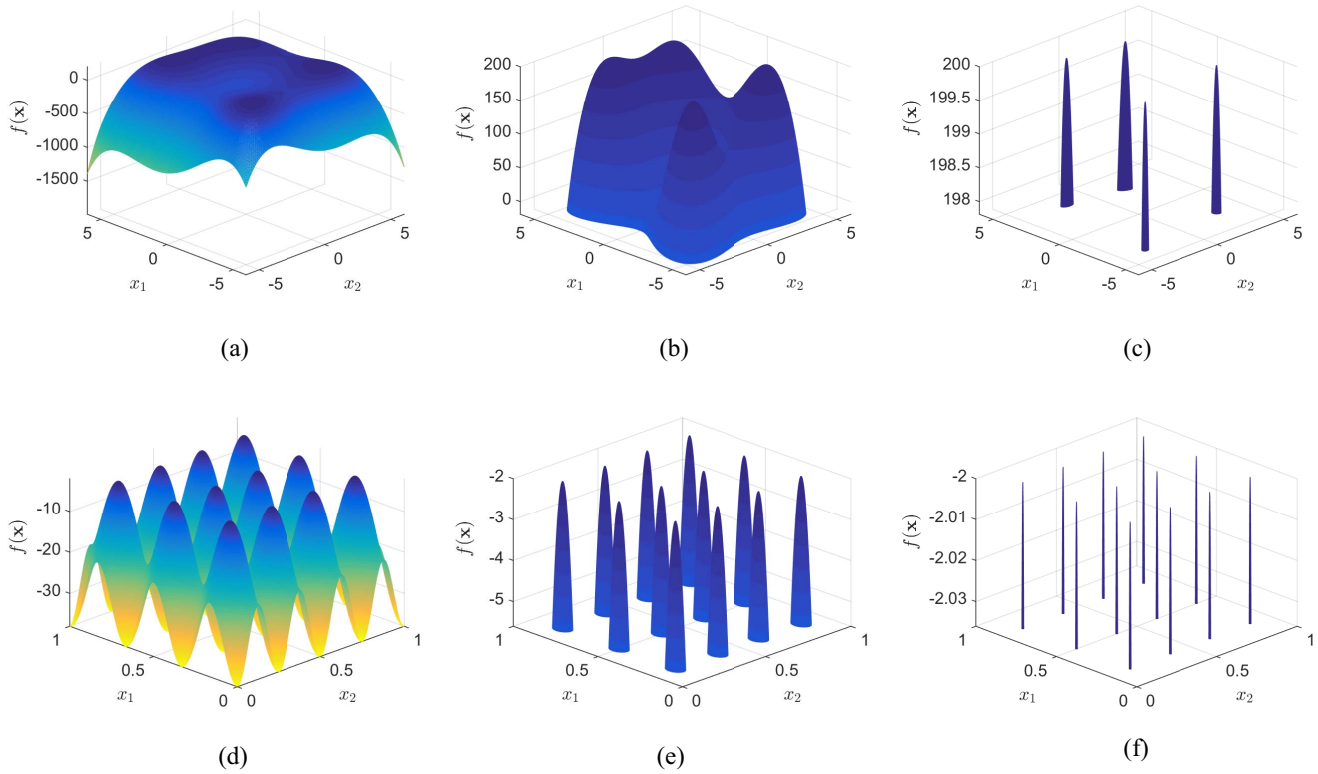


Fig. 3. Illustration to show that cutting different ratios on the same fitness landscape will result in different observations of peaks. $F4$ and $F10$ are two multimodal functions taken from the IEEE CEC 2013 benchmark test suite for MMO [56], which have a number of 4 and 12 optimal solutions (i.e., peaks of the same maximum height), respectively. (a) Entire fitness landscape of $F4$. Peaks of $F4$ obtained by cutting top (b) 10% of the fitness landscape and (c) 0.1% of the fitness landscape. (d) Entire fitness landscape of $F10$. Peaks of $F10$ obtained by cutting top (e) 10% of the fitness landscape and (f) 0.1% of the fitness landscape.

Algorithm 3 Binary Cutting-Based APD

Input: approximate fitness landscape $\mathcal{D} = (X, Y)$, parameter η to determine the initial cutting ratio;
Output: detected peak set $\mathcal{P}$;
/*Initial Cutting*/
$[y_{min}, y_{max}] \leftarrow$ extreme fitness values in Y;
$\mathcal{D}_c = \{\forall(x_i, y_i) \in \mathcal{D} : y_i > (y_{max} - \eta(y_{max} - y_{min}))\}$;
/*Binary Cuttings*/
while $\mathcal{D}_c \neq \emptyset$ **do**
  $\mathcal{P} = \mathcal{P} \cup APD(\mathcal{D}_c)$; // Algorithm 4
  /*Cutting Top 50% of $\mathcal{D}_c$*/
  $y_{min} \leftarrow$ minimal fitness value in Y_p;
  $\mathcal{D}_c = \{\forall(x_i, y_i) \in \mathcal{D}_c : y_i > \frac{y_{min}+y_{max}}{2}\}$;
end while

Algorithm 4 APD

Input: cutting slice of approximate fitness landscape $\mathcal{D}_c = (X_c, Y_c)$, where $X_c = (\mathbf{x}_{c,1}, \mathbf{x}_{c,2}, ...)$;
Output: detected peak set $\mathcal{P}_c$;
$k = 0$;
while $\mathcal{D}_c \neq \emptyset$ **do**
  $k = k + 1$;
  /*Detecting the k-th Peak in $\mathcal{D}_c$*/
  $\sigma = \max_i\{\min_{j\neq i}\|\mathbf{x}_{c,i} - \mathbf{x}_{c,j}\|_1\}$; // adaptive threshold to determine whether two data points are connected
  $\psi_k = \{(\mathbf{x}_\sigma, \mathbf{y}_\sigma)\}$; // initializing peak set ψ_k with the data point having the neighboring distance equal to σ
  for $i = 1$ to $|\psi_k|$ **do**
    $\mathcal{D}_c = \mathcal{D}_c \backslash \{(\mathbf{x}_i, \mathbf{y}_i)\}$;
    $I_{con} = \{\forall j \in \{1, ..., |\mathcal{D}_c|\} : \|\mathbf{x}_i - \mathbf{x}_j\|_1 \leq \sigma\}$; // data points connected to $\mathbf{x}_i$ in the decision space
    $\psi_k = \psi_k \cup \mathcal{D}_c(I_{con})$; // adding all connected data points to peak set ψ_k
  end for
  $\mathcal{P}_c = \mathcal{P}_c \cup \{\psi_k\}$;
end while

although there exist four peaks in this fitness landscape, due to the mild gradients around the optimal solutions, the peaks are almost invisible. By contrast, for the fitness landscape shown in Fig. 3(d), the 12 peaks can be clearly observed due to the sharp gradients. Therefore, in order to automatically locate the peaks for any given MMOP, we propose a binary cutting-based APD method.

The motivation of the proposed peak detection method is based on the observation that by cutting the top of a multimodal fitness landscape, peaks will become disconnected to each other due to the gaps thus generated between them, as illustrated in Fig. 3(c) and (f). In this way, the peak detection problem is equivalently transformed to a graph connectivity

Algorithm 5 Local Search

1: **Input:** detected peak set $\mathcal{P} = \{\psi_1, \psi_2, ..., \psi_{|\mathcal{P}|}\}$, the MMOP to be optimized $g(\mathbf{x})$;
2: **Output:** optimal solution set $\mathcal{S}$;
3: **for** $k = 1$ to $|\mathcal{P}|$ **do**
4: /*Extracting Seed Solution*/
5: $(\mathbf{x}_0, \mathbf{y}_0) \leftarrow$ solution with the best fitness in peak ψ_k;
6: /*Performing Local Search*/
7: $(\mathbf{x}_k^*, \mathbf{y}_k^*) \leftarrow LocalOptimizer((\mathbf{x}_0, \mathbf{y}_0), g(\mathbf{x}))$;
8: $\mathcal{S} = \mathcal{S} \cup \{(\mathbf{x}_k^*, \mathbf{y}_k^*)\}$;
9: **end for**

detection problem, and each peak can be seen as a maximal connected subgraph, where decision vectors are connected inside the same peak but disconnected to those in any other peaks. Such cutting-based techniques performed on archived approximate fitness landscapes are commonly seen in the field of traditional global optimization [57]–[59]. Moreover, considering that the same cutting ratio applied to different fitness landscapes can generate completely different peaks, where as an example, the peaks in Fig. 3(e) are isolated but those in Fig. 3(b) are still fully connected, we propose a binary cutting strategy which is performed on the top of an approximate fitness landscape, such that peaks inside different cutting slices can be iteratively detected.

As summarized in Algorithm 3, the proposed binary cutting-based APD method begins with an initial cutting performed on top of the approximate fitness landscape $\mathcal{D}$, thus generating the initial cutting slice $\mathcal{D}_p$, where the cutting ratio is specified by a parameter $\eta \in (0, 1)$. Afterwards, binary cuttings are iteratively performed on the basis of $\mathcal{D}_p$, where in each iteration, the peaks inside the cutting slice $\mathcal{D}_p$ are detected successively using the APD as presented in Algorithm 4. In the detection of each peak, a threshold σ is adaptively calculated (step 7) to determine whether neighboring data points belong to the same peak, without introducing any additional parameters. The above procedure, as presented from lines 4 to 16, is iteratively operated until all data points in the cutting slice $\mathcal{D}_p$ are allocated to a corresponding peak, thus $\mathcal{D}_p$ becoming empty.

It is worth noting that the binary cutting-based APD method bases the assumption that there are only a finite number of optimal solutions such that the peaks are isolated in different subregions of the fitness landscape. However, it is interesting to see that the method is still able to detect a number of peaks even if an MMOP has an infinite number of continuously distributed optimal solutions, where the detailed discussion can be found in Section V-C.

D. Local Search

Once the peak set $\mathcal{P}$ is obtained using Algorithms 3 and 4, independent local search can be performed inside each peak using an existing single-objective optimizer. In the case that a DM is only interested in part of the peaks, he/she can choose to perform local search on specific peaks according to personal preferences; while if there are no specific DM's preferences available, general optimization can be performed on each peak successively, as presented in Algorithm 5.

To begin with, the data point with the best fitness value is first extracted as a seed solution. Afterwards, local search can be performed by merging the seed solution into the initial population. It should be noted that, since the local search is merely performed inside a decision space region specified by a given peak, we suggest that the search space should be constrained to a small hyperbox around the seed solution, where each dimension is set as 5% of feasible range as defined by (2). Besides, since there is no specific requirement for the local optimizer, in practice, any single-objective optimizer that has reliable exploitation capability is applicable.

IV. Experimental Study

In order to assess the performance of the proposed EMO-MMO,[2] a series of experiments are conducted on the IEEE CEC 2013 benchmark test suite for MMO[3] (CEC 2013 test suite for short hereafter) [56]. The CEC 2013 test suite consists of 20 functions in total, as summarized in Section I in the supplementary material, where $F1$–$F10$ are widely adopted test functions in the MMO community, and $F11$–$F20$ are some composition functions.

To begin with, some general comparisons are made between the proposed EMO-MMO and three state-of-the-art algorithms for MMO, namely, MOMMOP [44], NMMSO [60], and NEA2 [61], where MOMMOP is a recently proposed multimodal algorithm based on EMO techniques, and NMMSO and NEA2 are the winning entries of the IEEE CEC 2015 and IEEE CEC 2013 competitions for MMO, respectively. Moreover, performance of the proposed MOFLA method and the peak detection method is further assessed using some illustrative case studies. Finally, the sensitivity analysis of the allocation of FEs is conducted.

A. Benchmark Comparisons

1) Experimental Settings: For fair comparisons, all experimental settings are as recommended in [56]. Each algorithm is run for 50 independent times, and the termination condition for each test function is the maximum number of FEs as summarized in Section I in the supplementary material. For the three compared algorithms, namely, MOMMOP [44], NMMSO [60], and NEA2 [61], we adopt the parameter settings as suggested in their respective original publications. Given D as the number of decision variables, the specific settings of each algorithm are summarized as follows: for MOMMOP, the population size settings are listed in Section I in the supplementary material, and the parameter scaling factor is set to $\eta = 40D(t/t_{\max})$, where t and $t_{\max}$ are the current number and maximum number of FEs, respectively; for NMMSO, the single swarm size is set to $N = 10D$, and the maximum number of swarms to increment is set to

[2]Source code of EMO-MMO can be downloaded from: https://github.com/ranchengcn/EMO-MMO.

[3]Source code of the CEC 2013 test suite can be downloaded from: https://github.com/mikeagn/CEC2013.

TABLE I
PARAMETER SETTINGS FOR EACH COMPONENT OF EMO-MMO. FOR THE LOCAL SEARCH COMPONENT, IN ADDITION TO THE SWARM SIZE $m = 20$, THE OTHER PARAMETER φ IS SET TO 0 AS SUGGESTED IN [62]

Components of EMO-MMO	Parameter Settings
MOFLA	$N = 500$
Peak Detection	$\eta = 0.1$
Local Search (CSO [62])	$m = 20$, $\varphi = 0$

max_inc = 100; and for NEA2, the population size is set to $40D$.

In contrast to the problem-dependent population sizing of the three compared algorithms, the proposed EMO-MMO adopts a consistent population size of 500. Besides, the initial cutting ratio in Algorithm 3, as a control parameter to be specified in EMO-MMO, is set to $\eta = 0.1$ for all test functions, and some further discussion on the settings of η are given in Section IV-C. To quickly setup the local search as presented in Algorithm 5, we directly apply the recently proposed competitive swarm optimizer (CSO) [62] as the local optimizer without any modification. As the final solution set, the candidate solutions obtained by Algorithm 5 are merged into the final population obtained by Algorithm 2. To be clear, the parameter settings for each component of EMO-MMO are summarized in Table I.

It is worth noting that, since both MOFLA component (Algorithm 2) and local search component (Algorithm 5) require a certain number of FEs to work properly, we allocate 50% of the maximum FEs to each component, respectively, without any bias. Further discussion on the allocation of FEs can be found in Section IV-D.

2) Performance Measurements: To evaluate the results obtained by each algorithm,[4] the two measurements as recommended in [56] are used as performance indicators, namely, the peak ratio (PR)

$$\mathrm{PR} = \frac{\sum_{\mathrm{run}=1}^{\mathrm{NR}} \mathrm{NPF}_i}{\mathrm{NKP} \times \mathrm{NR}} \tag{17}$$

and the success rate (SR)

$$\mathrm{SR} = \frac{\mathrm{NSR}}{\mathrm{NR}} \tag{18}$$

where NR denotes the total number of runs, NPF_i denotes the number of global optima found in the ith run, and NKP and NSR are the number of known global optima and the number of successful runs, respectively. As the threshold for the calculation of SR and NR, the accuracy level ε, which indicates the tolerable difference of function values between the true global optimal solutions and the candidate solutions, should be specified. Correspondingly, three accuracy levels of $\varepsilon = 10^{-1}$, $\varepsilon = 10^{-3}$, and $\varepsilon = 10^{-5}$ are used in the experiments.

3) Experimental Results: In general, EMO-MMO shows most competitive performance in comparison with MOMMOP, NMMSO, and NEA2, having achieved 100% SR on 12 out of 20 functions at all accuracy levels. To be specific, we have the following observations.

[4]Source code of the performance measurements can be downloaded from: https://github.com/mikeagn/CEC2013.

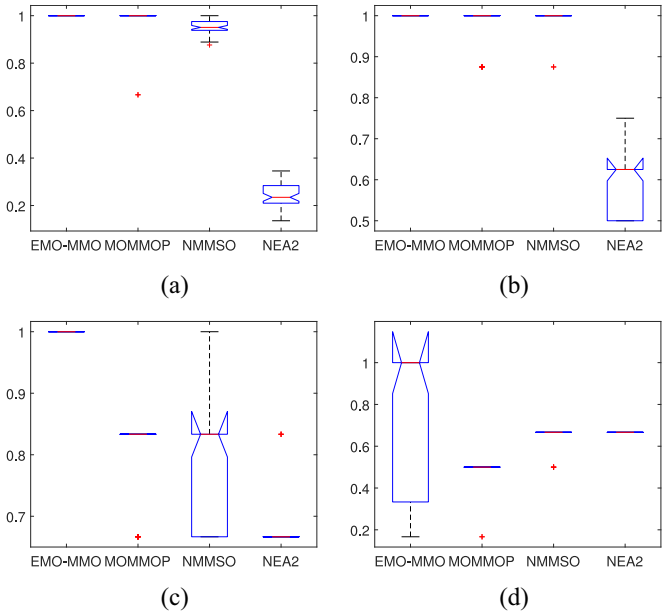


Fig. 4. Boxplots of the results obtained by each algorithm in 50 runs at accuracy level $\varepsilon = 10^{-1}$. (a) $F8$. (b) $F12$. (c) $F14$. (d) $F18$.

All of the four algorithms have shown promising performance at all accuracy levels on $F1$–$F5$, which have a relatively small number of global optima. The only exception is that NEA2 has failed to find all the global optimal at accuracy level of $\varepsilon = 10^{-5}$ on $F4$, which has a very smooth fitness landscape as shown in Fig. 3(a). By contrast, the proposed EMO-MMO, which is based on an APD method, has managed to locate all of the four global optima at all accuracy levels. For $F6$–$F9$, which have 18, 36, 81, and 216 global optima, respectively, both EMO-MMO and MOMMOP have also achieved high PR values. This observation indicates that the proposed EMO-MMO is capable of handling MMOPs with a large number of global optima. By contrast, NEA2 is significantly outperformed by the other three algorithms, especially on $F8$ and $F9$, where NEA2 has only achieved around 20% and 60% PR, respectively.

While the fitness landscapes of $F1$–$F10$ are relatively simpler, the remaining ten functions, $F11$–$F20$, are composition functions which have more complex fitness landscapes. As a consequence, EMO-MMO is the only algorithm that is still able to achieve 100% SR at all accuracy levels on part of them. By contrast, the other three algorithms have all failed to achieve 100% SR on all these function, especially on $F15$–$F20$, where the SR is 0% at all accuracy levels. In fact, obtaining all global optimal solutions (i.e., achieving a successful run) on high-dimensional test functions such as $F15$–$F20$ can be challenging for any existing MMO algorithms [4]. Since the candidate solutions are very sparsely distributed in the high-dimensional decision space, it is very likely that some of the global optimal solutions are undetected (or lost), thus leading to 0% SR. Another interesting observation is that NEA2 has significantly outperformed all the other three algorithms on $F16$–$F20$, showing promising scalability to the number of decision variables. This is mainly due to the

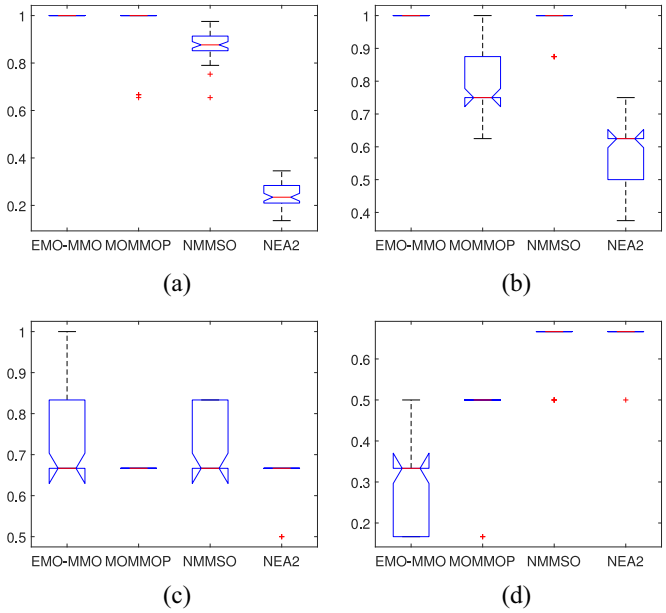


Fig. 5. Boxplots of the results obtained by each algorithm in 50 runs at accuracy level $\varepsilon = 10^{-5}$. (a) $F8$. (b) $F12$. (c) $F14$. (d) $F18$.

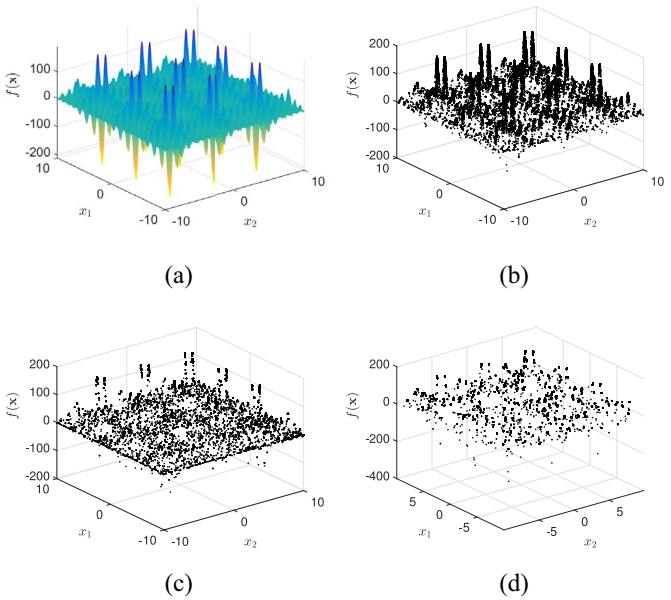


Fig. 6. Approximate fitness landscapes of $F6$ obtained by EMO-MMO, MOMMOP, and NMMSO using 40 000 FEs. (a) True fitness landscape. Approximate fitness landscape obtained by (b) MOFLA, (c) MOMMOP, and (d) NMMSO.

effectiveness of the nearest-better clustering method adopted in NEA2 [3], which is designed to enhance the performance of the algorithm on both low-dimensional and high-dimensional problems. As will be presented in Section V-B, the scalability of the proposed EMO-MMO can be also potentially improved by adopting a specially tailored reproduction operator.

For further observations, boxplots of the results obtained by each algorithm on each test function in 50 runs are given in Section III in the supplementary material. Representatively, Figs. 4 and 5 show the boxplots of $F8$, $F12$, $F14$, and $F18$, where $F8$ has a large number of 81 global optimal solutions, $F12$ and $F14$ are low-dimensional composite functions which have complicated fitness landscapes, and $F18$ is the high-dimensional (10-D) instance of $F14$. One one hand, EMO-MMO shows generally robust performance at the low accuracy level of $\varepsilon = 10^{-1}$. On the other hand, at the higher accuracy level of $\varepsilon = 10^{-5}$, EMO-MMO still shows stable performance on $F8$ and $F12$, but its performance suffers from significant deterioration on $F18$. Besides, it is interesting to see that although NEA2 tends to occasionally lose some optimal solutions, its performance is quite stable regardless of the accuracy levels.

In summary, compared with MOMMOP, NMMSO, and NEA2, the proposed EMO-MMO has shown best performance on most test functions in the CEC 2013 test suite, with respect to both PR and SR. Since the performance of EMO-MMO is largely dependent on the proposed MOFLA method and the peak detection method, in the following sections, we present some empirical results to further demonstrate the advantages of both methods, especially when applied to preference-based decision-making.

B. Fitness Landscape Approximation

While most existing algorithms for MMO merely aim to find all optimal solutions, in practice, the DM may only be interested in some specific solutions of his/her preferences. In this scenario, achieving all optimal solutions can be quite inefficient, especially for problems with expensive FEs. To address such an issue, we demonstrate that, by consuming a certain number of FEs, the proposed EMO-MMO can be used to assist the decision-making process by obtaining an approximate fitness landscape together with adaptively detected peaks marked on it.

As an illustrative example, we have run the proposed MOFLA (Algorithm 2), MOMMOP and NMMSO for 40 000 FEs (only 20% of the maximum FEs as used in benchmark comparisons) on $F6$, and a large archive is used to record all the candidate solutions obtained by each algorithm as an approximation to the fitness landscape. As presented in Fig. 6, the approximate fitness landscapes obtained by MOFLA, MOMMOP and NMMSO show significantly different qualities. To be specific, MOFLA has obtained the best approximation to the fitness landscape, where the shapes of the sharp peaks are clearly visible; by contrast, for MOMMOP and NMMSO, most points are merely located on the top of the peaks.

As the most important subcomponent in MOFLA, the proposed grid-based diversity indicator $d_{\text{grid}}(\mathbf{x})$ is crucial to the performance of the whole algorithm. To assess the effectiveness of $d_{\text{grid}}(\mathbf{x})$, we have performed further empirical comparisons between it and the classic Euclidean distance diversity indicator $d_3(\mathbf{x})$ as given in (9). To be specific, we use $F10$, which a relatively simple fitness landscape [as shown in Fig. 3(d)], to conduct the experiments. As evidenced in Fig. 7, MOFLA has completely failed the approximation to the fitness landscape of $F10$ once $d_{\text{grid}}(\mathbf{x})$ is replaced with $d_3(\mathbf{x})$, which confirms the effectiveness of the proposed grid-based diversity indicator $d_{\text{grid}}(\mathbf{x})$.

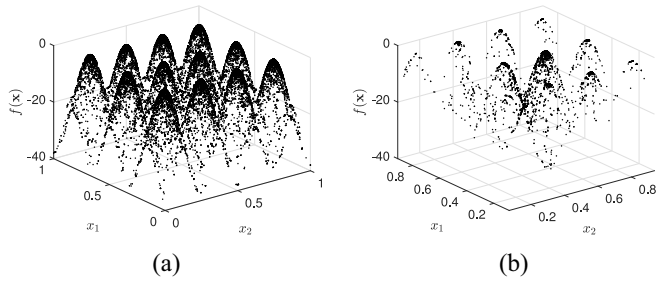


Fig. 7. Approximate fitness landscapes of $F10$ obtained by MOFLA with grid distance-based diversity ind $d_{\text{grid}}(\mathbf{x})$ and classic Euclidean distance-based $d_3(\mathbf{x})$ using 40 000 FEs. Approximate fitness landscape obtained by (a) MOFLA with $d_{\text{grid}}(\mathbf{x})$ and (b) MOFLA with $d_3(\mathbf{x})$.

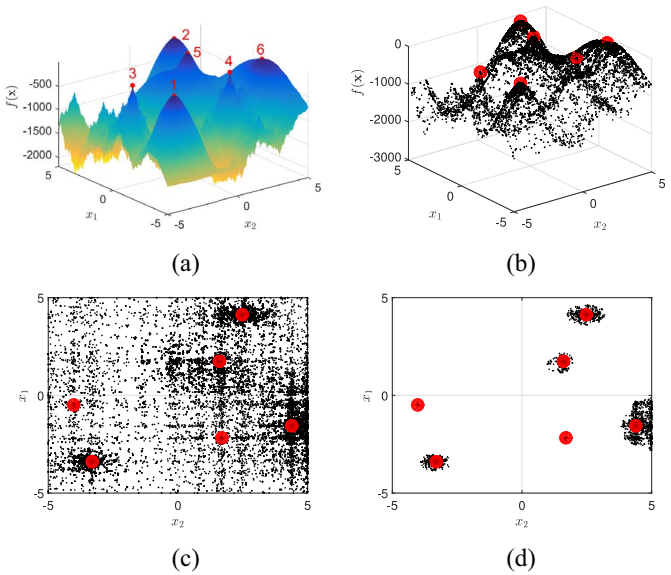


Fig. 8. Approximate fitness landscape marked with the detected peaks of $F11$ obtained by EMO-MMO. (a) True fitness landscape. (b) Approximate fitness landscape with detected peaks marked using solid dots. (c) Decision space of the approximate fitness landscape with detected peaks marked using solid dots. (d) Decision space of initial cutting slice with detected peaks marked using solid dots.

C. Peak Detection

On the basis of the approximate fitness landscapes obtained by Algorithm 2, we are able to further apply Algorithm 3 to the detection of peaks where optimal solutions may exist. For example, as shown in Fig. 8(a), although the six global optimal solutions of $F12$ have the same fitness, they are located on the peaks of significantly different landscapes. Considering the robustness in engineering designs, the DM may prefer to perform further local search inside the smooth peaks (e.g., peak 6), where the optimal solutions are less sensitive to the decision variable tunings than those on sharp peaks (e.g., peaks 3 or 4). Therefore, performing peak detection can be particularly meaningful in practical engineering designs.

As presented in Fig. 8(b), despite that some of the peaks are quite sharp while the others are more smooth, all of the six peaks in the approximate fitness landscape of $F11$ has been successfully detected, which indicates the robustness of the proposed adaptive strategy. In addition, as evidenced in

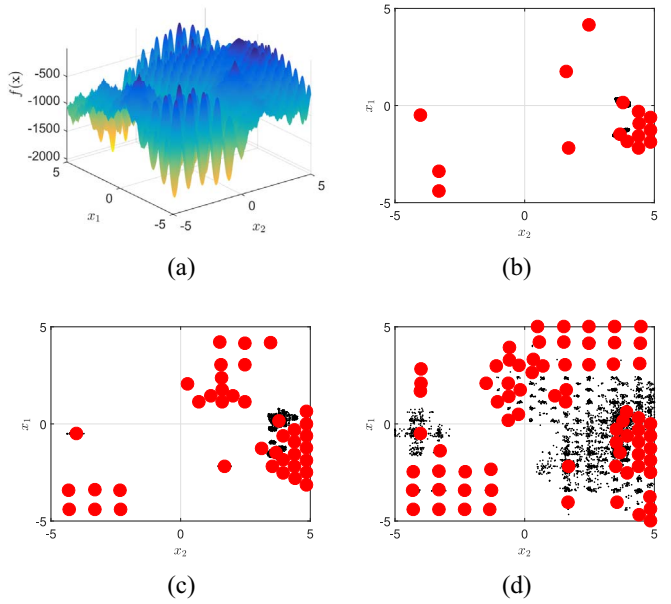


Fig. 9. Cuttings slices marked with detected peaks on $F12$ obtained by EMO-MMO using different settings of initial cutting ratio η. (a) True fitness landscape. Decision space of initial cutting slice with a number of (b) 16 detected peaks marked using solid dots ($\eta = 0.05$), (c) 40 detected peaks marked using solid dots ($\eta = 0.1$), and (d) 73 detected peaks marked using solid dots ($\eta = 0.3$).

Fig. 8(c), data points around the peaks show significantly higher density than those in other regions of the decision space, which indicates that the MOFLA method is able to adaptively adjust the distribution of the candidate solutions according to the specific locations of the peaks, thus avoiding useless explorations in the barren regions. Moreover, as presented in Fig. 8(d), cutting the fitness landscape to a certain slice will remove the sparsely distributed points which have poor fitness. Consequently, the DM is able to determine the ROIs (e.g., the region of peak 6) to perform further local search.

In addition to the sparsely distributed global optimal solutions such as in $F11$, for some other problems, there can also exist a large number of local optimal solutions. In this case, the number of peaks to be detected can be somehow controlled by the settings of the initial cutting ratio η. To further verify the robustness of the proposed peak detection method in terms of different settings of η, we conduct additional experiments using $F12$, which has a large number of local optimal solutions. As shown in Fig. 9, the proposed peak detection method has obtained different numbers of peaks with different settings of η, where the smaller η is set, the fewer peaks (with higher fitness) will be left in the cutting slice, and vice versa. Therefore, setting η to a too large value can lead to some potential issues. First, if the problem has a large number of local optimal solutions, a large initial cutting slice can cause a large number of local peaks to be detected, thus costing more FEs to exploit each of them in the local search procedure. Second, a large initial cutting slice may contain too many sample points, thus increasing the computational cost of the peak detection procedure. To avoid such issues, we suggest

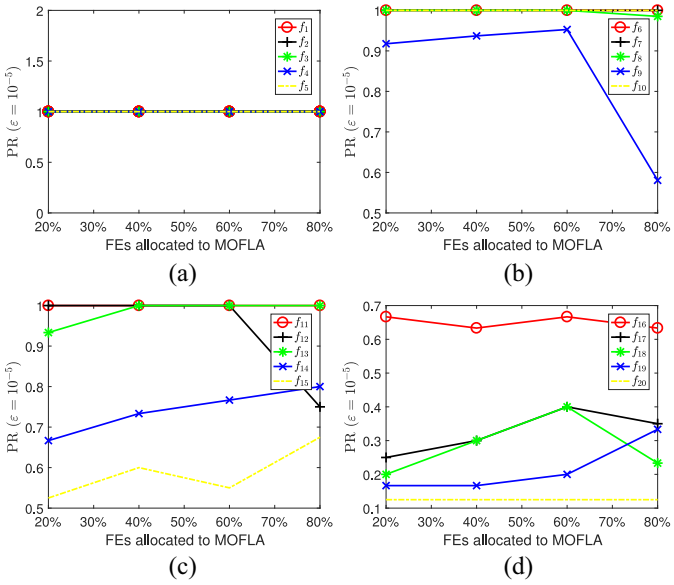


Fig. 10. Mean PR values obtained by EMO-MMO at accuracy level of $\varepsilon = 10^{-5}$ using different percentages of maximum FEs allocated to the MOFLA component. (a) f_1–f_5. (b) f_6–f_{10}. (c) f_{11}–f_{16}. (d) f_{17}–f_{20}.

that a small value η should always be considered, e.g., $\eta = 0.1$ as adopted in this paper.

D. Allocation of Fitness Evaluations

In the proposed EMO-MMO, both of the MOFLA component (Algorithm 2) and the local search component (Algorithm 5) require a certain number of FEs. In our benchmark studies, without any priori knowledge available, the two components are considered equally important to the black-box benchmark test functions, and thus 50% of the maximum FEs are allocated to each component, respectively. As further investigation, we have performed some sensitivity analysis on the allocation of FEs.

As indicated by the results summarized in Fig. 10, the performance of EMO-MMO is not particulary sensitive to the allocation of FEs on most test functions, except f_9 and f_{12}, which have a large number of global and local optimal solutions, respectively. Intuitively, this is due to the fact a larger number of detected peaks (i.e., potential optimal solutions) will require more FEs for the local search to be performed on each peak successively. In this case, allocating too many FEs to the MOFLA component will consequently result in insufficient FEs for local search, thus leading to poor performance of the algorithm. Therefore, in practice, the DM may allocate the FEs on the basis of approximate fitness landscape and according to personal preferences.

V. Discussion

A. Effectiveness of Grid Coordinate System

In the following, we elaborate some further discussion to demonstrate the advantages of the grid coordinate system over the real coordinate system in terms of diversity measurement for the proposed EMO-MMO. To begin with, we replace the grid-based normalization method in (10) with the following real-valued normalization method:

$$x'_{t,i,j} = \left(\frac{x_{t,i,j} - x^{\min}_{t,j}}{x^{\max}_{t,j} - x^{\min}_{t,j}} \right) \tag{19}$$

where $x'_{t,i,j}$ falls into range $[0, 1]$, such that the diversity indicator as formulated in (11)–(14) is calculated in the real coordinate space. With this real-valued normalization method, we conduct some experimental comparisons between the modified EMO-MMO (denoted as EMO-MMO-R for short hereafter) and the original EMO-MMO on the CEC 2013 benchmark test suite, where all the experimental settings remain the same as those adopted in Section IV.

As summarized by the results in Section IV in the supplementary material, EMO-MMO-R shows the same performance to EMO-MMO on simple test functions such as $F1$–$F6$, but is significantly outperformed by EMO-MMO on difficult test functions such as $F7$–$F9$ or $F11$–$F20$, which either have a large number of global optimal solutions or have a complicated composite fitness landscape. This is due to the fact that the real coordinate system fails to well balance between convergence and diversity in the decision space, thus causing the loss of part of the solution sets. Such empirical observations indicate that the proposed grid coordinate system is crucial to the performance of EMO-MMO in terms of diversity measurement, especially on those hard problems.

B. Reproduction Operator in MOFLA

For simplicity, the MOFLA method in this paper has been designed on the basis of the original NSGA-II, where the reproduction operator is the classic simulated binary crossover (SBX) operator plus the polynomial mutation operator (step 7 in Algorithm 2). As one of the most important components in an MOEA, the reproduction operator could substantially determine the search behaviors of the algorithm, thus influencing the performance of the proposed MOFLA. To this end, we conduct some further investigations by proposing a localized DE operator (refer to Section VI in the supplementary material) to replace the SBX operator in Algorithm 2, and rerun the modified EMO-MMO (denoted as EMO-MMO-DE for short hereafter) on the CEC 2013 test suite using the same settings as introduced in Section I.

As summarized in Section IV in the supplementary material, EMO-MMO-DE and original EMO-MMO have achieved the same (or very close) performance on simple low-dimensional test functions such as $F1$–$F6$ or $F10$–$F16$, while their performance is substantially different on difficult test functions such as $F7$–$F9$ or $F17$–$F20$. To be specific, the original EMO-MMO significantly outperforms EMO-MMO-DE on test functions $F7$–$F9$ which have a large number of global optimal solutions; by contrast, EMO-MMO-DE shows promising scalability on the high-dimensional test functions $F17$–$F20$, significantly outperforming the original EMO-MMO. Such observations indicate that the reproduction operator is crucial to the performance of the MOFLA method, hence, it is very likely that a specially tailored reproduction operator can

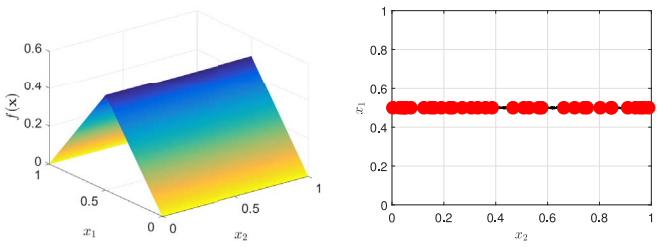


Fig. 11. Cuttings slices marked with detected peaks on the roof problem obtained by EMO-MMO. (a) True fitness landscape. (b) Decision space of initial cutting slice with detected peaks marked using solid dots.

improve the performance of EMO-MMO on specific problems of different types (e.g., high-dimensional problems).

C. Applicability to Infinite Optimal Set

As demonstrated by the experimental study in Section IV, the proposed EMO-MMO shows generally robust performance on a variety of test functions which have different numbers of optimal solutions. Although the number of optimal solutions varies from 1 to 216, all of the optimal solutions are still discretely distributed in the fitness landscapes. In practice, however, there may exist some problems where the optimal solutions are continuously distributed, thus leading to an infinite optimal set. To further investigate the performance of EMO-MMO on such kind of problems, we have specially designed a new test function, called a *roof problem*

$$f(x_1, x_2) = \begin{cases} x_1 & \text{if } x_1 \leq 0.5 \\ 1 - x_1 & 0.5 < x_1 \leq 1 \end{cases} \tag{20}$$

where $0 \leq x_1, x_2 \leq 1$. As shown in Fig. 11(a), this problem has an infinite global optimal set along the *roof ridge* defined by $x_1 = 0.5$.

In order to approximate the fitness landscape of the roof problem and detect the peaks where optimal solutions could exist, we run EMO-MMO for 50 000 FEs. As shown in Fig. 11(b), consequently, EMO-MMO has obtained a certain number of well distributed peaks along the "roof ridge," which implies the potential applicability of EMO-MMO to the problems having infinite optimal sets. Nevertheless, there are still some open issues worthy of further investigations. For example, compared to the dense distribution of the sampled candidate solutions in the optimal region, the distribution of the detected peaks is relatively sparse, and the exact number of detected peaks is not controllable. Besides, since EMO-MMO performs stochastic search behaviors, it also does not guarantee which exact peaks to be detected in each independent run. In this case, the DM may have to specify some ROIs in order to obtain solutions according to personal preferences, thus calling for the development of specially tailored preference integration/articulation methods.

VI. Conclusion

By taking advantage of EMO techniques in population diversity preservation, we have proposed an EMO-MMO algorithm. The proposed EMO-MMO first obtains an approximate fitness landscape marked with adaptively detected peaks, and then, local search is performed inside each peak independently. Our experimental results have demonstrated that the proposed EMO-MMO not only shows promising performance in the benchmark comparisons with some state-of-the-art algorithms, but also has good potential in assisting preference-based decision-makings in MMO.

While most existing MMO algorithms try to find all optimal solutions during one single run, the proposed EMO-MMO has adopted a two-stage framework: 1) to approximate the fitness landscape and 2) to exploit the ROIs. Technically, the framework has been designed to be flexible. For example, in the MOFLA component, both diversity indicator and reproduction operator are replaceable. Besides, the local search operator could also be any single-objective optimizer. Even the peak detection method could also be replaced as long as it is able to detect the ROIs (e.g., the peaks) on the basis of the approximate fitness landscape. In the future, we would like to investigate how to design new methods or operators to tackle more challenging (e.g., high-dimensional) MMOPs using such a framework. In addition, the visualization of high-dimensional multimodal landscapes is also worth investigating [63].

References

[1] S. Das, S. Maity, B.-Y. Qu, and P. N. Suganthan, "Real-parameter evolutionary multimodal optimization—A survey of the state-of-the-art," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 71–88, Jun. 2011.

[2] K.-C. Wong, "Evolutionary multimodal optimization: A short survey," *CoRR*, vol. abs/1508.00457, 2015. [Online]. Available: http://arxiv.org/abs/1508.00457

[3] M. Preuss, *Multimodal Optimization by Means of Evolutionary Algorithms*. Cham, Switzerland: Springer, 2015.

[4] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking multiple solutions: An updated survey on niching methods and their applications," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 518–538, Aug. 2017.

[5] K. Deb and S. Gulati, "Design of truss-structures for minimum weight using genetic algorithms," *Finite Elements Anal. Design*, vol. 37, no. 5, pp. 447–465, 2001.

[6] M. Preuss, P. Burelli, and G. N. Yannakakis, "Diversified virtual camera composition," in *Proc. Eur. Conf. Appl. Evol. Comput.*, 2012, pp. 265–274.

[7] M. Kronfeld, A. Dräger, M. Aschoff, and A. Zell, "On the benefits of multimodal optimization for metabolic network modeling," in *Proc. GCB*, 2009, pp. 191–200.

[8] O. M. Shir, C. Siedschlag, T. Bäck, and M. J. J. Vrakking, "Niching in evolution strategies and its application to laser pulse shaping," in *Proc. Int. Conf. Artif. Evol.*, Lille, France, 2005, pp. 85–96.

[9] E. Pérez, M. Posada, and F. Herrera, "Analysis of new niching genetic algorithms for finding multiple solutions in the job shop scheduling," *J. Intell. Manuf.*, vol. 23, no. 3, pp. 341–356, 2012.

[10] E. Pérez, M. Posada, and A. Lorenzana, "Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms," *Soft Comput.*, vol. 20, no. 5, pp. 1879–1896, 2016.

[11] D.-X. Chang, X.-D. Zhang, C.-W. Zheng, and D.-M. Zhang, "A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem," *Pattern Recognit.*, vol. 43, no. 4, pp. 1346–1360, Apr. 2010.

[12] S. Kamyab and M. Eftekhari, "Feature selection using multimodal optimization techniques," *Neurocomputing*, vol. 171, pp. 586–597, Jan. 2016.

[13] C. Castillo, G. Nitschke, and A. Engelbrecht, "Niche particle swarm optimization for neural network ensembles," in *Proc. Eur. Conf. Artif. Life*, Budapest, Hungary, 2009, pp. 399–407.

[14] G. Guanqi and Y. Shouyi, "Evolutionary parallel local search for function optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 864–876, Dec. 2003.

[15] O. M. Shir, "Niching in evolutionary algorithms," in *Handbook of Natural Computing*. Heidelberg, Germany: Springer, 2012, pp. 1035–1069.

[16] A. Pétrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proc. IEEE Int. Conf. Evol. Comput.*, Nagoya, Japan, 1996, pp. 798–803.

[17] G. Singh and K. Deb, "Comparison of multi-modal optimization algorithms based on evolutionary algorithms," in *Proc. Annu. Conf. Genet. Evol. Comput.*, Seattle, WA, USA, 2006, pp. 1305–1312.

[18] O. J. Mengshoel and D. E. Goldberg, "Probabilistic crowding: Deterministic crowding with probabilistic replacement," in *Proc. Genet. Evol. Comput. Conf.*, 1999, pp. 409–416.

[19] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2. Portland, OR, USA, 2004, pp. 1382–1389.

[20] P. Darwen and X. Yao, "Every niching method has its niche: Fitness sharing and implicit sharing compared," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, Berlin, Germany, 1996, pp. 398–407.

[21] D. E. Goldberg and L. Wang, "Adaptive niching via coevolutionary sharing," Gen. Algorithms Evol. Strategy Eng. Comput. Sci., Univ. Illinois at UrbanaÚChampaign, Champaign, IL, USA, Tech. Rep. 97007, 1997, pp. 21–38. [Online]. Available: https://pdfs.semanticscholar.org/f872/d929d49a2a5837a1361e543c41246532422d.pdf

[22] B. Sareni and L. Krahenbuhl, "Fitness sharing and niching methods revisited," *IEEE Trans. Evol. Comput.*, vol. 2, no. 3, pp. 97–106, Sep. 1998.

[23] X. Yin and N. Germay, "A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization," in *Artificial Neural Nets and Genetic Algorithms*. Vienna, Austria: Springer, 1993, pp. 450–457.

[24] F. Streichert, G. Stein, H. Ulmer, and A. Zell, "A clustering based niching EA for multimodal search spaces," in *Proc. Int. Conf. Artif. Evol. (Evol. Artificielle)*, Marseille, France, 2003, pp. 293–304.

[25] G. R. Harik, "Finding multimodal solutions using restricted tournament selection," in *Proc. ICGA*, Pittsburgh, PA, USA, 1995, pp. 24–31.

[26] B.-Y. Qu and P. N. Suganthan, "Novel multimodal problems and differential evolution with ensemble of restricted tournament selection," in *Proc. IEEE Congr. Evol. Comput.*, Barcelona, Spain, 2010, pp. 1–7.

[27] P. J. Darwen and X. Yao, "Speciation as automatic categorical modularization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 2, pp. 101–108, Jul. 1997.

[28] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 207–234, 2002.

[29] J. Yao, N. Kharma, and P. Grogono, "Bi-objective multipopulation genetic algorithm for multimodal function optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 80–102, Feb. 2010.

[30] C. Stoean, M. Preuss, R. Stoean, and D. Dumitrescu, "Multimodal optimization by means of a topological species conservation algorithm," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 842–864, Dec. 2010.

[31] B.-Y. Qu, P. N. Suganthan, and J.-J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.

[32] W. Gao, G. G. Yen, and S. Liu, "A cluster-based differential evolution with self-adaptive strategy for multimodal optimization," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1314–1327, Aug. 2014.

[33] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. New York, NY, USA: Springer, 2011, pp. 760–766.

[34] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[35] B.-Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, Jun. 2013.

[36] J. E. Fieldsend, "Multi-modal optimisation using a localised surrogates assisted evolutionary algorithm," in *Proc. UK Workshop Comput. Intell.*, Guildford, U.K., 2013, pp. 88–95.

[37] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 246–263, Apr. 2015.

[38] Q. Yang *et al.*, "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 191–205, Apr. 2017.

[39] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12. New York, NY, USA: Springer, 1999.

[40] S. Wessing, M. Preuss, and G. Rudolph, "Niching by multiobjectivization with neighbor information: Trade-offs and benefits," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 103–110.

[41] K. Deb and A. Saha, "Multimodal optimization using a bi-objective evolutionary algorithm," *Evol. Comput.*, vol. 20, no. 1, pp. 27–62, 2012.

[42] S. Bandaru and K. Deb, "A parameterless-niching-assisted bi-objective approach to multimodal optimization," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 95–102.

[43] A. Basak, S. Das, and K. C. Tan, "Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 666–685, Oct. 2013.

[44] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, Apr. 2015.

[45] K. Deb, "Multi-objective evolutionary algorithms," in *Springer Handbook of Computational Intelligence*. Heidelberg, Germany: Springer, 2015, pp. 995–1015.

[46] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.

[47] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 838–856, Dec. 2015.

[48] J. D. Knowles and D. W. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 100–116, Apr. 2003.

[49] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation," *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 253–274, Jun. 2003.

[50] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, 2005.

[51] L. Rachmawati and D. Srinivasan, "Dynamic resizing for grid-based archiving in evolutionary multi objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Singapore, 2007, pp. 3975–3982.

[52] M. Li, J. Zheng, K. Li, Q. Yuan, and R. Shen, "Enhancing diversity for average ranking method in evolutionary many-objective optimization," in *Proc. Int. Conf. Parallel Problem Solving Nat. (PPSN)*, 2010, pp. 647–656.

[53] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, Oct. 2013.

[54] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[55] L. Li and K. Tang, "History-based topological speciation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 136–150, Feb. 2015.

[56] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization," Evol. Comput. Mach. Learn. Group, RMIT Univ., Melbourne, VIC, Australia, Tech. Rep., 2013. [Online]. Available: http://goanna.cs.rmit.edu.au/~xiaodong/cec13-lsgo/competition/cec2013-lsgo-benchmark-tech-report.pdf

[57] R. Becker and G. Lago, "A global optimization algorithm," in *Proc. 8th Allerton Conf. Circuits Syst. Theory*, 1970, pp. 3–12.

[58] A. H. G. R. Kan, C. G. E. Boender, and G. T. Timmer, "A stochastic approach to global optimization," in *Computational Mathematical Programming*. Heidelberg, Germany: Springer, 1985, pp. 281–308.

[59] A. Törn and S. Viitanen, "Topographical global optimization," in *Recent Advances in Global Optimization*. Princeton, NJ, USA: Princeton Univ. Press, 1992, pp. 384–398.

[60] J. E. Fieldsend, "Running up those hills: Multi-modal search with the niching migratory multi-swarm optimiser," in *Proc. IEEE Congr. Evol. Comput.*, Beijing, China, 2014, pp. 2593–2600.

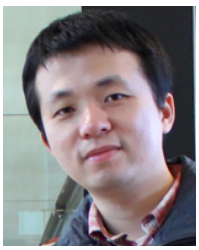
[61] M. Preuss, "Improved topological niching for real-valued global optimization," in *Proc. Eur. Conf. Appl. Evol. Comput.*, 2012, pp. 386–395.

[62] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015.
[63] R. Cheng, M. Li, and X. Yao, "Parallel peaks: A visualization method for benchmark studies of multimodal optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 263–270.

Ran Cheng (M'16) received the B.Sc. degree in computer science and technology from Northeastern University, Shenyang, China, in 2010 and the Ph.D. degree in computer science from the University of Surrey, Guildford, U.K., in 2016.

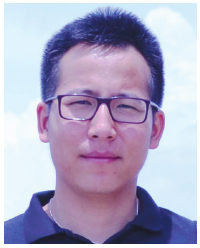
He is currently a Research Fellow with the CERCIA Group, School of Computer Science, University of Birmingham, Birmingham, U.K. His current research interests include evolutionary multiobjective optimization, evolutionary multimodal optimization, model-based evolutionary algorithms, large-scale optimization, and swarm intelligence.

Dr. Cheng is a recipient of the 2018 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award. He is the Founding Chair of the IEEE Symposium on Model Based Evolutionary Algorithms.



Miqing Li received the Ph.D. degree in computer science from the Department of Computer Science, Brunel University London, London, U.K., in 2015.

He is currently a Research Fellow with CERCIA, School of Computer Science, University of Birmingham, Birmingham, U.K. His current research interests include evolutionary multiobjective and many-objective optimization, covering algorithm design, performance assessment, test problem construction and visualization, and their application in diverse fields.



Ke Li (M'17) received the B.Sc. and M.Sc. degrees in computer science and technology from Xiangtan University, Xiangtan, China, in 2007 and 2010, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2014.

He was a Post-Doctoral Research Associate with Michigan State University, East Lansing, MI, USA, and a Research Fellow with the University of Birmingham, Birmingham, U.K. He is currently a Lecturer (Assistant Professor) in Data Analytics with the Department of Computer Science, University of Exeter, Exeter, U.K. He has served as the regular reviewer and has published several research papers in renowned journals, such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS. His current research interests include evolutionary multiobjective optimization, large-scale optimization, and machine learning and applications in water engineering and software engineering.

Xin Yao (M'91–SM'96–F'03) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.Sc. degree from the North China Institute of Computing Technology, Beijing, China, in 1985, and the Ph.D. degree from USTC in 1990.

He is currently a Chair Professor of computer science with the Southern University of Science and Technology, Shenzhen, China, and a part-time Professor of computer science with the University of Birmingham, Birmingham, U.K. His current research interests include evolutionary computation, ensemble learning, and their applications in software engineering.

Dr. Yao was a recipient of the 2001 IEEE Donald G. Fink Prize Paper Award, the 2010, 2016, and 2017 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Awards, the 2010 BT Gordon Radley Award for Best Author of Innovation (Finalist), the 2011 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, the Royal Society Wolfson Research Merit Award in 2012, the IEEE Computational Intelligence Society (CIS) Evolutionary Computation Pioneer Award in 2013, and several other best paper awards. He was the President of the IEEE CIS from 2014 to 2015, and the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2003 to 2008. He is a Distinguished Lecturer of the IEEE CIS.